INFORMAL COMMUNICATION: Please do not put in the file

Interview Agenda:

- Discussion of exemplary differences between the application/claims and the cited references; and
- Discussion of proposed amendments

Exemplary Differences

Independent Claim 1

[0002] It does not appear that the cited references disclose the following (in particular, the emphasized text) from claim 1:

> capturing a list of events directed at composing a video frame during processing of a set of commands by a graphics processing unit, ***the events comprising commands sent to the graphics processing unit from a memory location***;

[0003] Rather, the combination of the cited references, and specifically, Brown, merely teach capturing graphics call sequences and generating an optimized graphics call sequence that provides for the same rendering as the original graphics call sequence. See column 11, lines 40-48. Brown recites the redundant, conflicting, or otherwise unnecessary graphics states calls are eliminated to reduce the total number of graphics state calls. See column 13, lines 60-65. The cited art, and specifically, Brown, does not recite capturing commands sent to the GPU from memory. The remaining cited art does not overcome the deficiencies of Brown. It is further believed the above analysis applies to independent claim 18 as well.

INFORMAL COMMUNICATION: Please do not put in the file

Independent Claim 9 and Dependent Claim

[0004] It does not appear that the cited references disclose the following (in particular, the emphasized text) from claim 9:

> capture the state of the graphics processing unit for each of the captured events, ***the state comprising transitory states of internal variables of the graphics processing unit;***

[0005] As mentioned above, the combination of the cited references, and specifically, Brown, merely teach capturing graphics call sequences and generating an optimized graphics call sequence that provides for the same rendering as the original graphics call sequence. See column 11, lines 40-48. Brown recites the redundant, conflicting, or otherwise unnecessary graphics states calls are eliminated to reduce the total number of graphics state calls. See column 13, lines 60-65. The cited art, and specifically Brown, does not recite the state of the GPU also includes transitory states of internal variables. The remaining cited art does not overcome the deficiencies of Brown.

Proposed Amendments

[0006] Please see the attached Appendix of Proposed Claim Amendments. I would like to discuss your opinion regarding the proposed amendments in light of the currently cited references.

INFORMAL COMMUNICATION: Please do not put in the file

[0007] Thank you in advance for scheduling time for this interview. I look forward to discussing this with you.

Respectfully Submitted,

Dated: October 29, 2008

By:_____

Michael D. Carter
Reg. No. 56661
(512) 505-8164
michaelcarter@leehayes.com
www.leehayes.com

My Assistant: Carly Taylor
(512) 505-8162
carly@leehayes.com

INFORMAL COMMUNICATION: Please do not put in the file

Appendix of Claims with Proposed Amendments

1. (Currently Amended) A method for controlling presentation of information to facilitate performance analysis for processing, the method comprising:

capturing a list of events directed at composing a video frame during processing of a set of commands by a graphics processing unit, the events comprising commands sent to the graphics processing unit from a memory location;

capturing the state of the graphics processing unit for each of the captured events;

displaying a listing of the captured events as well as information regarding the processing of the events;

displaying a window including a video frame portion that displays a rendering of the video frame;

receiving a user selection of one of the events of the listing;

modifying the selected event;

setting the state of the graphics processing unit to the captured state associated with the selected event;

executing the selected event in the graphics processing unit; and

displaying in the video frame portion a visual representation of the frame resulting from the execution of the selected event.

2. (Original) A method as recited in claim 1, wherein the information regarding the processing of the events comprises a value representing how long it took for processing of the events by a processing unit to finish.

INFORMAL COMMUNICATION: Please do not put in the file

3. (Original) A method as recited in claim 1, further comprising:

displaying a timeline portion including a timeline having a plurality of bars, each bar corresponding to a particular one of the events, wherein a location of each bar on the timeline indicates when the corresponding event occurred relative to the other events during processing.

4. (Original) A method as recited in claim 1, further comprising:

allowing the user to select a warning window, to be displayed as the frame portion, wherein the warning window identifies violations of one or more recommendations for programming a processing unit that processed the set of commands.

5. (Cancelled).

6. (Previously Presented) A method as recited in claim1, further comprising:

allowing the user to select one of multiple views to be displayed in the frame portion, wherein the multiple views include a render target view that shows the frame as it is drawn at different chronological points while being drawn, a depth buffer view that shows a depth value for each pixel in the frame at different chronological points while the frame is being drawn, and a wireframe view that shows an outline of each triangle rendered in the frame at different chronological points while the frame is being drawn.

7. (Previously Presented) A method as recited in claim 1, further comprising:

allowing the user to select one of multiple views to be displayed in the frame portion, wherein the multiple views include an overdraw view that shows graphically how many times each pixel in the frame is drawn, and a fill rate view that shows how fast a graphics processing unit that drew the frame was running when each pixel in the frame was drawn.

8. (Previously Presented) A method as recited in claim 1, further comprising:

a debugger portion that identifies a pixel shader program or vertex shader program that was executed by a graphics processing unit in drawing the frame, and further identifies input and output register values for each instruction in the shader program as it executed in drawing the selected pixel.

9. (Currently Amended) One or more computer readable media having one or more instructions that, when executed by one or more processors, causes the one or more processors to:

capture a list of events directed at composing a video frame during processing of a set of commands by a graphics processing unit;

capture the state of the graphics processing unit for each of the captured events, the state comprising transitory states of internal variables of the graphics processing unit;

display a first window that identifies the list of events that have been captured during the drawing of the video frame;

receive a user selection of one of the events in the list;

modify the selected event;

set the state of the graphics processing unit to the captured stated associated with the selected event;

execute the selected event in the graphics processing unit; and

display a second window including a video frame portion that shows how the frame appears at different points while being drawn.

10. (Original) One or more computer readable media as recited in claim 9, wherein the first window further identifies, for each of the identified events, a value representing how long it took for drawing of the event by a graphics processing unit to occur.

11. (Original) One or more computer readable media as recited in claim 9, wherein the one or more instructions further causes the one or more processors to:

display a third window including a timeline having a plurality of bars, each bar corresponding to a particular one of the identified events, wherein a location of each bar on the timeline indicates when the corresponding event occurred relative to the other events during drawing of the frame.

12. (Previously Presented) One or more computer readable media as recited in claim 9, wherein the one or more instructions further causes the one or more processors to:

allow the user to select one of multiple views to be displayed in the second window, wherein the multiple views include a render target view that shows the frame as it is drawn at different points while being drawn, a depth buffer view that shows a depth value for each pixel in the frame at different points while the frame is being drawn, and a wireframe view that shows an outline of each triangle rendered in the frame at different points while the frame is being drawn.

INFORMAL COMMUNICATION: Please do not put in the file

13. (Original) One or more computer readable media as recited in claim 9, wherein the one or more instructions further causes the one or more processors to:

allow the user to select one of multiple views to be displayed in the second window, wherein the multiple views include an overdraw view that shows graphically how many times each pixel in the frame is drawn, and a fill rate view that shows how fast a graphics processing unit that drew the frame was running when each pixel in the frame was drawn.

14. (Original) One or more computer readable media as recited in claim 9, wherein the one or more instructions further causes the one or more processors to:

allow the user to select a warning window, to be displayed as the second window, wherein the warning window identifies violations of one or more recommendations for programming a graphics processing unit that drew the frame.

15. (Original) One or more computer readable media as recited in claim 9, wherein the one or more instructions further causes the one or more processors to:

display a pixel history window that identifies each of the events that affects a user-selected pixel of the frame.

16. (Original) One or more computer readable media as recited in claim 9, wherein the one or more instructions further causes the one or more processors to:

INFORMAL COMMUNICATION: Please do not put in the file

display a debugger window that identifies a pixel shader program or vertex shader program that was executed by a graphics processing unit in drawing the frame, and further identifies input and output register values for each instruction in the shader program as it executed in drawing the selected pixel.

17. (Original) One or more computer readable media as recited in claim 9, wherein the one or more instructions further causes the one or more processors to:

display a mesh debugger window that includes information about a single mesh of the frame, and a table that shows the attribute values for each vertex in the mesh as well as an output of a vertex shader program for each vertex in the mesh.

18. (Currently Amended) A system comprising:

a memory;

a processor coupled to the memory; and

a plurality of instructions stored in the memory and executed by the processor to present a user interface to enable a user to view information regarding a frame of video, the user interface comprising:

a list of events including events captured during composition of a video frame by a graphics processing unit, wherein each captured event represents a command submitted to the graphics processing unit, the events comprising commands sent to the graphics processing unit from a memory location;

an events window that identifies the events captured during the composition of the video frame;

a frame window that shows how the video frame appeared immediately after a particular event was finished being drawn.

19. (Original) A system as recited in claim 18, wherein the graphics processing unit is part of another device coupled to the system.

20. (Original) A system as recited in claim 18, wherein the user interface further comprises:

a timeline window including a timeline having a plurality of bars, each bar corresponding to a particular one of the identified events, wherein a location of each bar on the timeline indicates when the corresponding event occurred relative to the other events during drawing of the frame.

21. (New) A method as recited in claim 1, wherein capturing the list of events further comprises obtaining data from the memory location.

22. (New) A method as recited in claim 1, wherein capturing the list of events further comprises determining if the memory location was previously referenced by a differing command.

23. (New) One or more computer readable media as recited in claim 9, wherein the internal variables of the graphics processing unit are stored in an internal register.

INFORMAL COMMUNICATION: Please do not put in the file

24. (New) A method as recited in claim 18, wherein the list of events captured further comprises obtaining data from the memory location.

25. (New) A method as recited in claim 18, wherein the list of events captured further comprises determining if the memory location was previously referenced by a differing command.